IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Application No.: 10/824,099	§		
Filed: April 14, 2004	§	Examiner:	Won, Michael Young
Inventors:	§	Group/Art Unit:	2155
Ajay Kumar	§	Atty. Dkt. No:	5681-54100
Title: State Data Persistence in a Distributed Computing Environment	§		

PRE-APPEAL BRIEF REQUEST FOR REVIEW

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

Applicants request review of the final rejection in the above-identified application. Claims 1-19 and 24-29 are pending in the application. The Examiner rejected claims 1-6, 9-19 and 24-29 under 35 U.S.C. § 102(e) as being anticipated by Mullins (U.S. Patent 7,103,600), and claims 7 and 8 under 35 U.S.C. § 103(a) as being unpatentable over Mullins in view of Jacobs et al. (U.S. Patent 6,385,643) (hereinafter "Jacobs"). Applicants note the following **clear errors** in Examiner's rejection.

Independent claims 1 and 9:

1. Mullins does not anticipate a server cluster comprising a plurality of server nodes.

The Examiner refers to Mullins, col. 1, line 67 to col. 2, line 3 as teaching this aspect of Applicant's claim. However, the cited portion of Mullins belongs to the "Background of the Invention" section of Mullins, and describes systems for accessing data stores from object oriented languages [col. 1, lines 30-31]. In a rejection based on § 102, it is improper to rely on this portion of Mullins that is clearly not part of the same system of Mullins that the Examiner relies upon elsewhere. Moreover, the cited lines actually recount problems with persisting data to such data stores, noting that the data object model "may be distributed over multiple physical computer machine locations or even distributed over multiple Internet website locations that may be independent of the data stores." There is no mention whatsoever of a server cluster including multiple server nodes. In the Advisory Action, the Examiner cites Mullins at col. 1, 2, and 16 as teaching this aspect of Applicant's claim. However, Mullins merely refers to data objects that "may logically span multiple relational tables or multiple object databases, and may even be distributed over a logical (or hypothetical) computer system involving multiple physically independent computer systems or even multiple website locations." Having data objects distributed over multiple

physical computers does not imply having a server cluster having multiple server nodes. No one of ordinary skill in the art would ever consider the teachings of Mullins to anticipate a server cluster.

2. Mullins does not anticipate a server container within a server node of a multi-node server cluster, and one or more applications configured to execute within that server container.

The Examiner refers to Mullins, col. 16, lines 49-53, which refers to a programming module of the CocoNavigator API [see col. 15, lines 13-16; col. 18, lines 37-45] which can be configured to operate as a tool to create, access, support and correctly manage a CDOG (complex data object graph) navigation model in a server environment, and to persist any changes to the CDOG navigation model when the navigation model is distributed across a local network or when the navigation model involves a distributed network (e.g., a navigation model distributed across internet connections). The programming modules of the CocoNavigator API are programming tools providing support to an object computer language programmer for manipulating complex data object graphs [col. 18, lines 37-45; col. 12, lines 19-38]. They are not applications configured to execute within a server container of a server node in a multi-node server cluster. In the Advisory Action, Examiner asserts that Mullins teaches an application executing on a server container, citing "the CocoNavigator API, or an associated computer program module, can be configured to operate as a tool to create, access, support and correctly manage a CDOG navigation model in a server environment (e.g., in an EJB container) and to persist any changes to the CDOG navigation model when a navigation model is distributed across a local network or when the navigation model involves a distributed network (e.g., a navigation model distributed across internet connections)." However, contrary to the Examiner's assertion, Mullins does not describe that the CocoNavigator API, or an associated tool for creating, accessing, supporting, and managing the CDOG navigation model, runs within a server container inside a server node of a multi-node server cluster. Mullins clearly teach that the CocoNavigator API operates as a programming tool, not as a cluster server application executing within each server container of each server node of a cluster.

3. Mullins does not anticipate a JDO persistence manager configured to detect changes to application state data within the server container and to persist the application state data.

The Examiner refers to Mullins, col. 8, lines 8-12, as teaching this aspect of Applicant's claim. However, the cited portion of Mullins describes a programming object which includes programming code for communicating with a persistence manager API (application programming interface) to persist the object, its data, and/or links to other objects. Mullins is not at all relevant to a persistence manager detecting changes to *application state data* within a server container *for an application executing* within

that server container, or to persisting that *application state data*. In the Advisory Action, the Examiner refers to Mullins at col. 7. Here Mullins simply recites "persisting any changes to an instance of the CDOG model." The Examiner asserts that "changes in the CDOG model result in a change in the state." However, the software component described by Mullins in the cited paragraph accesses a repository and persists actions such as "creating, maintaining, accessing, navigating, updating or deleting complex data objects." This is clearly not application state data within a server for an application executing within that server container, and persisting that application state data. The Examiner is ignoring the specific limitations of the claim.

4. Mullins does not anticipate a persistent data store coupled to the cluster, configured to store application state data of the applications executing within each respective server container, and configured to make the application state data accessible to each of the plurality of server nodes.

The Examiner refers to Mullins, col. 4, lines 8-18, as teaching this aspect of Applicant's claim. However, the cited portion of Mullins simply defines the term "data object," in the context of the document of Mullins, as an object which holds data, "and is likely to have its contents stored in a persistent data source of a computer system." Mullins makes no mention whatsoever of a persistent data store coupled to a cluster, nor of storing application state data of the applications executing within each respective server container, and making the application state data accessible to each of the cluster server nodes. In the Advisory Action, Examiner refers to the same cited portions of Mullins mentioned previously, but makes no new argument, and does not address Applicant's argument.

5. Mullins does not anticipate, wherein in response to detecting a change in application state data within the server container, the JDO persistence manager is configured to persist only a changed portion of the application state data within the respective server container to the persistent data store.

The Examiner refers to Mullins, col. 7, lines 24-34, and col. 35, lines 33-35, as teaching this aspect of Applicant's claim. However, the cited portion of Mullins at col. 7, lines 24-34 only teaches providing a computer software component which accesses a repository and persists actions such as "creating, maintaining, accessing, navigating, updating or deleting complex data objects," and which can persist a portion of a data object graph (a CDOG, representing one data object related to another data object), and which can persist changes to an instance or to the repository definition. Mullins does not teach a response to detecting a change in *application state data* within a server container, much less persisting only a changed portion of the *application state data* within the respective server container to

the persistent data store. The cited portion of Mullins at col. 35, lines 33-35, simply recites that Transaction objects persist only data object attributes that have changed. There is no indication, here or elsewhere in Mullins, of a response to detecting a change in *application state data* within the respective server container, much less persisting only a changed portion of the *application state data* within the respective server container to the persistent data store. In the Response to Arguments and Advisory Action, the Examiner does not address the substance of Applicant's arguments, and ignores the specific limitations of Applicant's claim.

Independent claims 14 and 24:

1. Mullins does not anticipate a Java Data Object (JDO) persistence manager detecting an access to application state data within a server, and in response to the detecting, determining whether the access alters the application state.

The Examiner refers to Mullins, col. 7, lines 14-23, col. 8, lines 8-12, and col. 10 lines 5-7, as teaching this aspect of Applicant's claim. However, Mullins teaches a "*user access* interface and a set of programming routines designed for creating or maintaining transparent persistence when *a user* is creating, maintaining, accessing and navigating complex data objects as a CDOG model [col. 10, lines 51-54, emphasis added]." Mullins at col. 7, lines 14-23 merely describes a computer software component which can access an object model repository in computer memory or in another temporary computer storage device, and persist the creating, maintaining, accessing, navigating, updating or deleting of complex data objects as a CDOG model. Mullins at col. 8, lines 8-12 only describes a programming object which includes programming code for communicating with a persistence manager API (application programming interface) to persist the object, its data, and/or links to other objects. And Mullins at col. 10 lines 5-7 is simply part of a description of Fig. 3, a programming flow chart representing the system for creating, displaying, updating, and persisting the data of a complex data object (CDO) which is part of a CDO graph. The cited portion of Mullins only refers to the first programming module of Fig. 3 and its displayable presentation page containing embedded object programming code. The displayable presentation page is presented to a user, and can display the data of a data object, or provide a user interface for creating or updating the data for that data object. The cited portion of Mullins only describes the embedded programming code of the displayable presentation page, which includes a programming reference link to an associated programming object which is connected to another programming module containing logic for detecting changes to object data or to a graph associated with the object.

Thus, the cited portions of Mullins actually teach a set of programming routines for creating or maintaining transparent persistence when a user is creating, maintaining, accessing and navigating

complex data objects as a CDOG model via a displayable presentation page. Neither singly nor collectively do the cited portions of Mullins anticipate, or even have any relevance to, a persistence manager detecting an access to application state data within a server, and *in response to the detecting*, determining whether the access alters the *application state*.

2. Mullins does not anticipate, in response to determining that the access alters the application state within the server, persisting only the elements of the application state that are changed by the access to a persistent store that makes the application state accessible to the server and to one or more other servers.

The Examiner refers to Mullins, col. 7, lines 24-34, and col. 35, lines 33-35. However, Mullins at col. 7, lines 24-34 teaches providing a computer software component able to persist all or a portion of a CDOG (complex data object graph) and to persist any changes to the repository definition for the CDOG model. Mullins at col. 35, lines 33-35, simply recites that Transaction objects persist only data object attributes that have changed. Clearly, there is no indication at all in Mullins of, in response to determining that the access alters the application state within the server, persisting only the elements of the application state that are changed by the access to a persistent store that makes the application state accessible to the server and to one or more other servers.

In light of the foregoing remarks, Applicants submit the application is in condition for allowance, and notice to that effect is respectfully requested. If any extension of time (under 37 C.F.R. § 1.136) is necessary to prevent the above referenced application from becoming abandoned, Applicant hereby petitions for such an extension. If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert & Goetzel PC Deposit Account No. 501505/5681-54100/RCK.

Respectfully submitted,
/Robert C. Kowert/
Robert C. Kowert, Reg. #39,255
Attorney for Applicants

Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C.
P.O. Box 398, Austin, TX 78767-0398
Phone: (512) 853-8850

Date: August 11, 2008